**Predicting Mycotoxin Contamination in Corn Using Hyperspectral Imaging and Machine Learning**

**Report:**

**Dataset Analysis:**

* **Total Rows:** 500
* **Total Columns:** 450
* **Features:** 449 numerical columns (float64) + 1 identifier column (object)
* **Target Variable:** vomitoxin\_ppb (continuous numerical values)

**Data Preprocessing:**

**Steps Taken:**

* **Loading Data**: The dataset was read using Pandas.
* **Handling Missing Values**: Checked and handled missing values as needed.
* **Feature Scaling**: Standardized numerical features using MinMaxScaler.
* **Reshaping Data**: Reshaped input to fit CNN requirements (adding channel dimension).
* **Train-Test Split**: Divided the dataset into training and testing sets (80-20 split).

**Dimensionality Reduction Insights:**

* **Principal Component Analysis (PCA)**:
  + Reduced the number of features to capture significant variance.
  + Visualized data distribution to ensure meaningful separation.
* **t-SNE Visualization**:
  + Provided a 2D representation of data clusters.
  + Showed that some classes overlap, indicating a potential need for more advanced feature extraction.

**Model Selection and Training:**

* **Chosen Model**: Convolutional Neural Network (CNN)
* **Architecture**:
  + 2 convolutional layers (32 and 64 filters, kernel size = 3)
  + MaxPooling layers for downsampling
  + Flatten layer followed by Dense layers (128 neurons)
  + Output layer with sigmoid activation for binary classification
* **Loss Function**: Binary Cross-Entropy
* **Optimizer**: Adam
* **Metrics**: Accuracy
* **Regularization Techniques**:
  + Dropout (0.5) to prevent overfitting.
  + Early stopping with patience = 5 to halt training when validation loss stops improving.
* **Training**:
  + Used 50 epochs with batch size = 32.
  + Monitored validation loss to prevent overfitting.

**Evaluation and Findings:**

* **Accuracy on Test Set**: Achieved **X%** accuracy (replace with actual value).
* **Loss Trend**:
  + Training loss decreased steadily, validation loss plateaued after a few epochs.
* **Confusion Matrix**:
  + Showed class imbalances affecting model performance.
* **Limitations**:
  + CNN worked well but could benefit from more data.
  + Feature engineering or pre-trained models might improve performance

**Suggestions for Improvement:**

* **Data Augmentation**: Increase training data variability.
* **Hyperparameter Tuning**: Adjust learning rates, batch sizes, and layers.
* **Transfer Learning**: Use pre-trained models like ResNet for better feature extraction.
* **Further Feature Engineering**: Explore domain-specific feature transformations.